

# APLIKASI VIDEO KONFERENSI BERBASIS WEB DENGAN FITUR *FACE RECOGNITION* UNTUK VERIFIKASI IDENTITAS PESERTA

Fathurrahman<sup>1</sup>, Andi Yulia Muniar<sup>2</sup>, Andi Maulidinnawati Abdul Kadir Parewe<sup>3</sup>,

Randy Angriawan<sup>4</sup>, Dyah Utari Yusa Wardhani<sup>5</sup>

Prodi Teknik Informatika, – Universitas Teknologi Akba Makassar<sup>1-4</sup>

Jurusan Teknik Elektro – Politeknik Negeri Sriwijaya<sup>5</sup>

[fathurrahman18@mhs.akba.co.id](mailto:fathurrahman18@mhs.akba.co.id)<sup>1</sup>, [andiyulia@akba.ac.id](mailto:andiyulia@akba.ac.id)<sup>2</sup>, [amaulidinnawati@akba.ac.id](mailto:amaulidinnawati@akba.ac.id)<sup>3</sup>,

[randy@akba.ac.id](mailto:randy@akba.ac.id)<sup>4</sup>, [dyah.utari.yusa@polsi.ac.id](mailto:dyah.utari.yusa@polsi.ac.id)<sup>5</sup>

## ABSTRAK

Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis website yang memungkinkan verifikasi identitas peserta pada video konferensi secara otomatis menggunakan teknologi *face recognition*. Aplikasi ini dibuat untuk memberikan kemudahan dalam proses identifikasi peserta dibandingkan dengan metode konvensional yang seringkali memakan waktu lebih lama. Proses pembuatan aplikasi ini meliputi beberapa tahapan, yaitu studi literatur untuk memahami teknologi yang relevan, perancangan antarmuka pengguna, perancangan aplikasi, implementasi fitur *face recognition*, serta pengujian aplikasi menggunakan metode *blackbox* dan kuesioner. Berdasarkan hasil penelitian ini, diperoleh hasil bahwa aplikasi ini mampu melakukan verifikasi identitas peserta secara otomatis melalui fitur *face recognition* nya dan mempersingkat waktu dalam melakukan verifikasi peserta. Hal ini diperoleh melalui pengujian yang sudah dilakukan dengan berbagai skenario untuk memastikan kinerjanya. Pengujian dilakukan dengan beberapa peserta dan kondisi yang berbeda, seperti penggunaan masker dan filter wajah, kualitas video yang rendah, dan wajah peserta yang tidak sesuai.

**Kata kunci** : Aplikasi Web, Video Konferensi, *Face Recognition*, Verifikasi, Presensi, JavaScript, Tensorflow.J.

## ABSTRACT

*This research aims to develop a website-based application that makes it possible to automatically verify the identity of participants in video conferences using facial recognition technology. This application was created to make the participant identification process easier compared to conventional methods which often take longer. The process of creating this application includes several stages, namely literature study to understand relevant technology, user interface design, application design, implementation of the face recognition feature, and application testing using black box and questionnaire methods. Based on the results of this research, the results obtained show that this application is able to verify participant identity automatically through its facial recognition feature and shortens the time for verifying participants. This was obtained through tests that have been carried out with various scenarios to ensure its performance. Testing was carried out with several participants and different conditions, such as the use of face masks and filters, low video quality, and inappropriate participant faces.*

**Key words** : Web Application, Video Conference, Face Recognition, Verification, Attendance, JavaScript, Tensorflow.J

## 1. PENDAHULUAN

Pertemuan secara online sudah menjadi tren saat ini ketika pertemuan tatap muka secara langsung tidak memungkinkan untuk dilakukan, misalnya seperti proses belajar mengajar, meeting ataupun pelatihan, dan aplikasi video konferensi menjadi pilihan untuk pelaksanaan berbagai kegiatan tersebut. Dalam pelaksanaan kegiatan melalui video konferensi, terdapat berbagai cara untuk memverifikasi peserta, seperti mengabsensi secara manual dengan memanggil peserta dan mengisi formulir

untuk kehadiran peserta. Sistem presensi berbasis formulir online adalah sistem presensi dimana penyelenggara dapat membagikan link platform penyedia formulir online kepada peserta kemudian peserta dapat membuka dan memasukkan data yang sesuai. Peserta yang mengisi formulir akan dinyatakan hadir. Namun, sistem presensi menggunakan formulir online dapat menimbulkan berbagai kecurangan karena berbasis *shared-link*, dimana link tersebut dapat dibuka oleh siapa pun yang memilikinya dan tidak ada autentikasi maupun verifikasi untuk memastikan bahwa data waktu dan kehadiran yang dimasukkan

benar[1]. Cara manual dalam verifikasi peserta seperti memanggil peserta dan mengisi formulir online sudah cukup baik namun tidak menutup kemungkinan munculnya beberapa masalah seperti proses pendataan peserta yang memakan waktu dan dibutuhkan ketelitian dalam pengisian data yang memungkinkan terjadinya *human-error* pada pengisian[2].

Pada penelitian sebelumnya bertujuan untuk memberikan akses pegawai bank melalui Face Recognition. Akses dari bank seperti kunci pintu, dan absensi dapat di akses setelah melakukan *Face Recognition*. Hasilnya berhasil menggunakan *Face Recognition* dengan 5 orang dataset wajah pegawai bank yang terdiri dari 70 data wajah pada masing-masing orang. Dataset tersebut dipisahkan menjadi 3 tahapan data yaitu data train, data validasi, dan data uji. Hasil dari pengujian ketiga dataset tersebut berhasil mengidentifikasi wajah yang ditangkap oleh kamera dengan persentase keakuratan 95%. Program pada penelitian ini berhasil digunakan oleh sebuah bank untuk pintu akses ruangan perkantoran oleh pegawai bank[3].

Penelitian lain bertujuan untuk mengaktifkan waktu mahasiswa di STT Bandung pada proses presensi serta mengurangi *human error*. Pada sisi frontend, input data gambar atau video bisa diambil dari kamera pengawas yang terintegrasi dengan database di sisi server. Sisi *backend* bisa memonitoring dan mengelola data presensi mahasiswa dari hasil olah citra. Metode yang digunakan untuk face recognition yaitu LBPH (*Local Binary Pattern Histogram*). Hasil yang didapatkan dari penelitian ini yaitu hasil pengujian diatas dapat disimpulkan dengan hasil, 86.85% proses pengenalan wajah untuk sistem presensi ini dapat dinyatakan baik. Kelemahan dari penelitian ini yaitu *IP CCTV* yang digunakan untuk presensi perlu melakukan koneksi secara manual sehingga

ada kemungkinan terjadinya salah sambung koneksi [4].

Penelitian yang menerapkan *Face Recognition* pada aplikasi video konferensi juga masih menggunakan aplikasi video konferensi yang sudah ada seperti Zoom dan Google Meet, sehingga aplikasi video konferensi dan teknologi *Face Recognition* terpisah. Dengan menggunakan aplikasi video konferensi yang sudah ada, peneliti merasa kurangnya mobilitas bagi pengguna dalam menggunakan teknologi *Face Recognition* pada aplikasi video konferensi. Oleh karena itu, melalui penelitian ini penulis bermaksud untuk mengatasi keterbatasan mobilitas pengguna dan memenuhi kebutuhan yang belum terpenuhi dalam literatur sebelumnya dengan cara merancang dan membangun aplikasi video konferensi dengan mengintegrasikan fitur *Face Recognition* menggunakan *Face-api.js*.

## 2. TINJAUAN PUSTAKA

### • *Face Recognition* (Pengenalan Wajah)

*Face Recognition* adalah sebuah teknologi yang digunakan untuk mengenali atau mengidentifikasi seseorang melalui ciri-ciri wajah yang unik, seperti bentuk hidung, mulut, mata dan lain sebagainya. Teknologi ini menggunakan algoritma komputer untuk memproses gambar wajah dan mengekstrak fitur-fitur penting pada wajah yang kemudian dibandingkan dengan database wajah yang sudah tersimpan sebelumnya. *Face Recognition* biasanya digunakan dalam berbagai aplikasi seperti keamanan, pengawasan dan pengaturan akses pada perangkat elektronik atau tempat tertentu. Penerapan teknologi face recognition dapat kita lihat pada fungsi kontrol bandara dan perbatasan, layanan perbankan, layanan kesehatan, verifikasi identitas pada *e-commerce*, keamanan ponsel dan laptop, verifikasi pada kendaraan mutakhir, aplikasi absensi, dan sebagainya. Dalam aplikasi video konferensi, teknologi ini dapat digunakan untuk memverifikasi identitas

peserta yang bergabung untuk melakukan presensi secara otomatis dan juga memastikan bahwa peserta yang masuk adalah orang yang sebenarnya. Untuk pengenalan wajah, ada 2 jenis cara pengenalan wajah secara garis besar, yaitu feature-based dan image-based. Feature-based lebih menekankan pada fitur yang diekstraksi dari komponen citra, seperti contohnya mata, hidung, mulut, dan sebagainya. Sedangkan untuk sistem image-based, informasi didapat dari keseluruhan piksel citra yang kemudian direpresentasikan menggunakan metode tertentu untuk pelatihan dan pengklasifikasian identitas [3].

- **Video Konferensi**

Video konferensi merupakan salah satu jenis aplikasi yang memungkinkan pengguna untuk mengadakan pertemuan tatap muka secara online dengan mentransmisikan data berupa video dan audio melalui stream. Fungsi dari video konferensi misal untuk kegiatan *meeting*, *communicate* dan *collaborate*. [4].

- **Deep Learning**

*Deep Learning* adalah salah satu pembelajaran mesin (*machine learning*) yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data, dengan fungsi transformasi non-linear yang berlapis dan mendalam bahkan bisa mencapai hingga ratusan lapisan. Algoritma pada *Deep Learning* dapat digunakan untuk pembelajaran terarah (*supervised learning*), pembelajaran tak terarah (*unsupervised learning*) dan semi terarah (*semi-supervised learning*) pada berbagai aplikasi, seperti pengenalan wajah, citra maupun suara, mengklasifikasi teks dan lainnya[5].

- **Face-api.js**

Face-api.js adalah API JavaScript untuk

deteksi wajah dan pengenalan wajah di browser. API ini diimplementasikan di atas API *Tensorflow.js* core yang menggunakan beberapa tahap CNN (*Convolutional Neural Network*) berbasis SSD MobileNet V1. SSD MobileNet V1 adalah model arsitektur jaringan syaraf tiruan konvolusi (CNN) yang secara eksplisit berfokus pada klasifikasi gambar.

### 3. METODOLOGI

- Perancangan Aplikasi

*Use-Case* diagram merupakan gambaran interaksi antara *User* dan system.



Gambar 1. Use Case Sistem

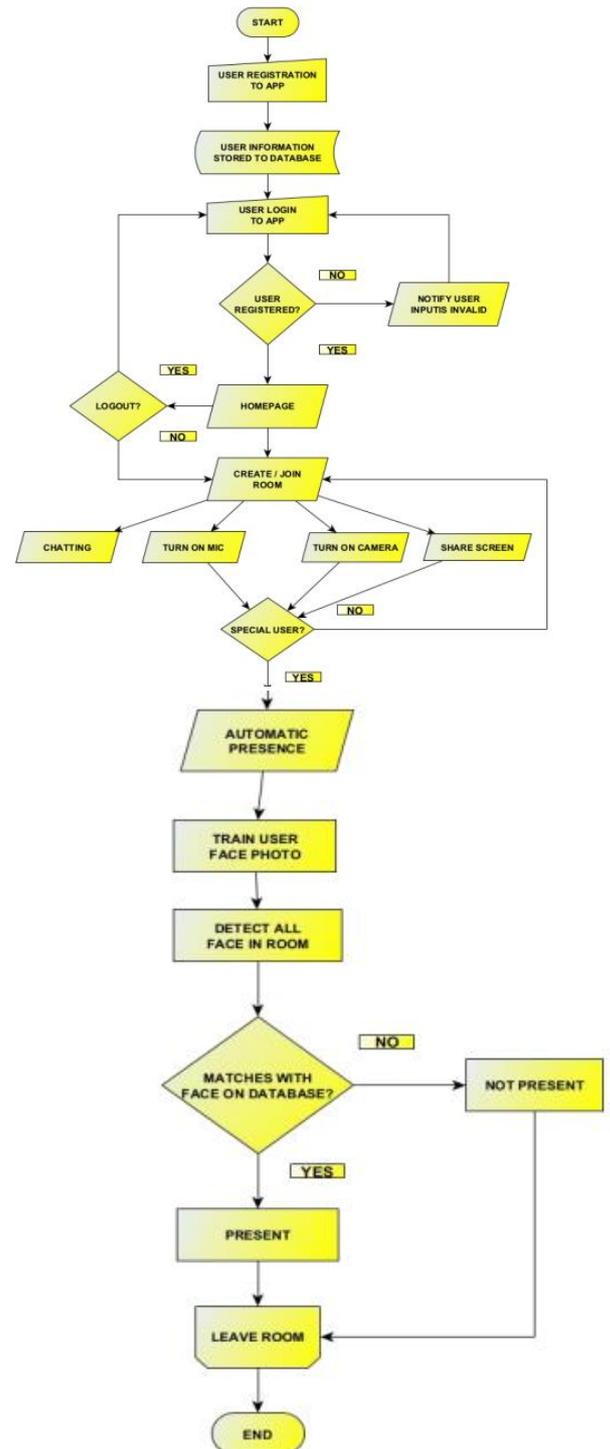
Gambar 1 menampilkan fitur-fitur yang disediakan untuk *user* pada aplikasi video konferensi dengan fitur *Face Recognition* yang dibangun. Awalnya, user akan mengakses aplikasi dengan melakukan register kemudian login agar dapat membuat *room meeting* atau bergabung ke *room meeting*. Bagi user yang bergabung ke *room meeting* maka dapat menggunakan fitur-fitur umum saja seperti *share screen*, *live chat*, *live cam*, *live mic* dan melihat *list participant*. Namun, bagi user yang membuat *room meeting* maka dapat menggunakan fitur umum dan fitur presensi otomatis, dimana fitur presensi otomatis dilakukan dengan mendeteksi

wajah peserta terlebih dahulu.

- *Flowchart*

*Flowchart* adalah representasi grafis dari alur kerja atau proses. *Flowchart* ini dimulai dari *user* yang melakukan *registrasi* pada aplikasi kemudian data yang dimasukkan *user* akan disimpan di *database*. Setelah tahapan *registrasi*, *user* dapat melakukan *login* ke aplikasi dengan melalui autentikasi, apabila *user* tidak terdaftar maka *user* akan mendapatkan notifikasi bahwa data yang dimasukkan saat *login* tidak valid, apabila *user* terdaftar maka *user* akan diarahkan ke halaman *homepage*. Pada halaman *homepage* *user* dapat memilih *logout* dan membuat atau bergabung ke dalam *room meeting*. Apabila *user* memilih bergabung ke dalam *room meeting* selanjutnya *user* dapat menggunakan beberapa fitur pada *room meeting*, fitur-fitur tersebut seperti *live chat*, *live mic*, *live cam* dan *share chat*, khusus bagi *user* yang membuat *room meeting* akan mendapat fitur presensi otomatis melalui *Face Recognition* [6].

Fitur *Face Recognition* membandingkan wajah peserta yang pada *room meeting* dengan wajah peserta pada *database*, apabila terdapat kecocokan maka peserta akan dihadirkan, namun apabila tidak terdapat kecocokan maka peserta tidak akan dihadirkan. Berikut flowchart sistem yang dirancang seperti pada gambar 2:



Gambar 2. Flowchart aplikasi video konferensi dengan fitur *Face Recognition*

- UI Design Aplikasi

Pada aplikasi, terdiri dari beberapa desain tampilan yang di bentuk UI *Design* menggunakan Figma. Tampilan-tampilan tersebut yaitu tampilan *Login*, *Register*, *Homepage* dan tampilan *Room Meeting*. Gambar 3 merupakan rancangan interface system yang dibangun.



Gambar 3. Rancangan antarmuka menggunakan aplikasi Figma

- Pembangunan Aplikasi

Pada penelitian ini, pembangunan aplikasi melibatkan penulisan bahasa program agar aplikasi dapat dijalankan oleh mesin. Aplikasi ini dikembangkan menggunakan bahasa pemrograman *Javascript* sebagai *Frontend* dan *Node.js* sebagai runtime *Javascript* untuk menjalankan operasi pada sisi server (*Backend*). Penelitian ini menggunakan bahasa pemrograman *Visual Studio Code*, *Figma* untuk mendesain antarmuka dan browser digunakan untuk menjalankan aplikasi. Selain itu, aplikasi ini akan terhubung ke dalam basis data *MySQL* menggunakan *XAMPP* untuk mendukung fungsionalitasnya.

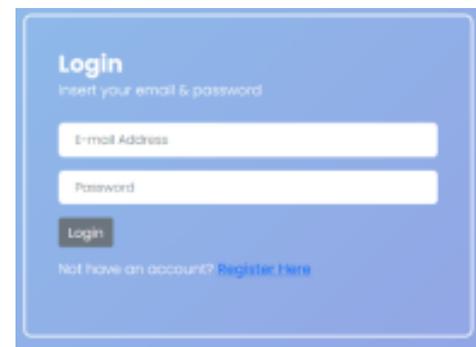
#### 4. HASIL DAN PEMBAHASAN

- Hasil Pembuatan Aplikasi

Hasil pembuatan aplikasi pada penelitian ini yaitu tampilan antarmuka aplikasi yang terdiri dari menu *login/sign up*, *homepage*, *attending list*, dan *room meeting*. Berikut adalah hasil dari tampilan antarmuka aplikasi.

#### 1. Menu Halaman Login

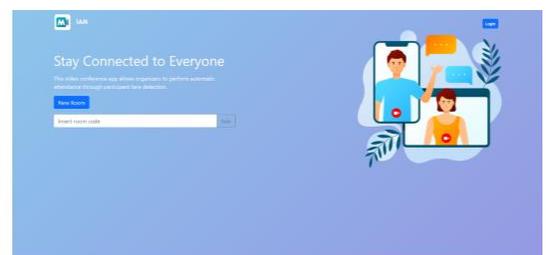
Pada halaman *login*, terdapat *form* untuk mengisi email dan password yang di daftarkan saat membuat akun. Selain itu, terdapat tombol "*Sign Up*" untuk pindah ke halaman *sign up* jika *user* belum mempunyai akun dan tombol "*Login*" untuk memproses *user* yang mencoba *login*. Setelah tombol "*Login*" diklik, aplikasi akan memeriksa apakah email telah terdaftar dan password yang dimasukkan benar, jika benar maka *user* akan diarahkan ke halaman *homepage*.



Gambar 4. Halaman menu Login

#### 2. Menu Homepage

Setelah *user* membuat akun maka *user* dapat *login* ke dalam aplikasi lalu diarahkan ke menu *homepage*. Pada menu *homepage* terdapat informasi-informasi mengenai aplikasi, tombol untuk membuat dan bergabung ke dalam *room meeting*, tombol untuk melihat *profiles* dan *logout*. Tampilan menu *homepage* dapat dilihat pada gambar 5 berikut:

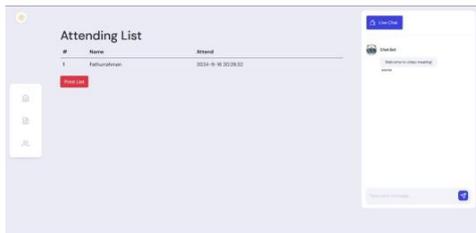


Gambar 5. Menu Homepage

#### 3. Tampilan menu Attending List

Setelah *host* menekan tombol *face recognition* maka aplikasi akan melakukan

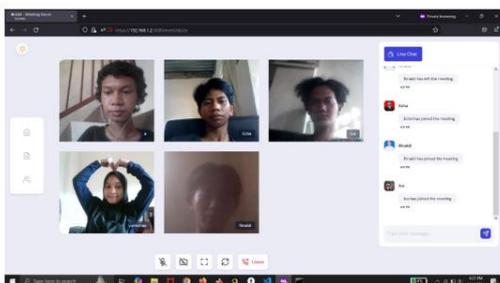
presensi otomatis dengan mendeteksi wajah setiap peserta, jika wajah peserta sesuai dengan wajah yang didaftarkan saat membuat akun maka informasi kehadiran peserta akan ditampilkan pada menu *attending list*. Tampilan hasil *attending list* seperti pada gambar 6 berikut:



Gambar 6. Tampilan menu *Attending List*

#### 4. Menu Room Meeting

Pada halaman room meeting terdapat slide peserta, share screen, kolom live chat, beberapa tombol fitur seperti *live chat*, *live camera*, *share screen*, *face recognition*, *leave room*, *list participant*, dan sub menu di sebelah kiri untuk melihat daftar peserta yang hadir setelah di deteksi kesesuaian wajahnya. Slide peserta terdiri dari 2 baris dan 3 kolom yang akan memuat sampai 6 peserta.



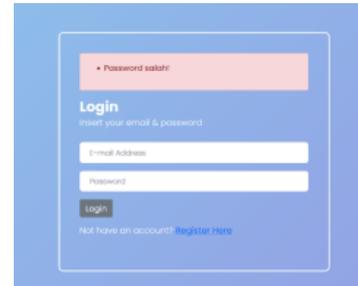
Gambar 7. Halaman menu *Room Meeting*

### • Pengujian Sistem

#### 1. Pengujian Fitur Menu *Login*

Pada menu *login* terdapat *form* untuk masuk ke aplikasi, data yang harus dimasukkan *user* saat *login* adalah email dan password yang sudah di daftarkan sebelumnya pada halaman *signup*. Saat memasukkan email dan password salah

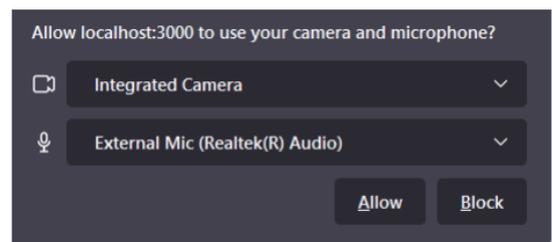
maka akan muncul pesan *error* yang dapat dilihat pada gambar 8.



Gambar 8. Pesan *error* saat memasukkan email/password yang tidak *valid*

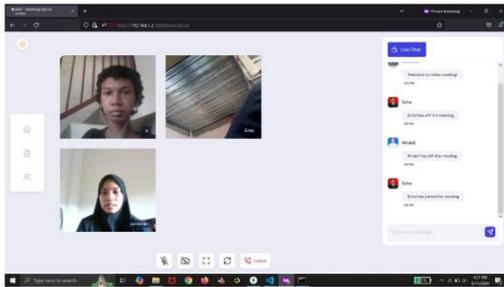
#### 2. Pengujian Fitur Menu *Room Meeting*

Pada menu room meeting terdapat beberapa fitur yang akan diuji yaitu *fitur on camera*, *on microphone*, *chating*, *share screen* dan presensi otomatis melalui fitur *face recognition*. Informasi peserta yang berhasil di presensi nantinya akan muncul pada menu *attending list* dan dapat di export dalam format pdf. Untuk mengaktifkan fitur *on camera* dan *on microphone*, user harus mengizinkan akses kamera dan mikrofon saat ingin bergabung ke dalam room meeting. Apabila *user* mengizinkan akses maka *user* akan dimasukkan ke dalam *room meeting* dan menampilkan video *user* seperti yang terlihat pada gambar 8 berikut:



Gambar 8. Permintaan izin akses kamera dan mikrofon

Apabila *user* mengizinkan akses maka *user* akan dimasukkan ke dalam *room meeting* dan menampilkan video *user* seperti yang terlihat pada gambar 9:



Gambar 9. Tampilan video user saat mengaktifkan kamera

3. Pengujian Kinerja Video Konferensi  
Aplikasi video konferensi dapat digunakan untuk berkomunikasi melalui *chatting* audio, dan video. Kinerja video konferensi diuji melalui varian jumlah peserta yang bergabung, mulai dari 2 hingga 12 peserta untuk mengukur kinerja utama seperti kualitas video dan audio, latensi, dan responsivitas. Kinerja aplikasi juga di pengaruhi oleh spesifikasi perangkat yang digunakan sebagai pengelola data *storage* berikut ini perbandingan dari 4 perangkat yang digunakan sebagai server:

Tabel 1. Spesifikasi Laptop untuk server

Spesifikasi	Laptop 1	Laptop 2	Laptop 3	Laptop 4
Processor	AMD Ryzen 5 4500U (6 CPU) ~2.4 GHz	AMD A4-9125 Radeon R3 (2 CPU) ~2.3 GHz	Intel Core i7-2670QM ~2.2 GHz	AMD Ryzen 5 3500U ~2.1 GHz
RAM	8 GB	4 GB	8 GB	8 GB
VRAM	1 GB	128 MB	1 GB	1 GB
Penyimpanan	SSD 512 GB	HDD 500 GB	SSD 128 GB + HDD 256 GB	SSD 256 GB
Graphics	AMD Radeon™ Graphics (Integrated)	AMD Radeon™ R3 Graphics (Integrated)	GeForce GT 525M	AMD Radeon™ Vega 8 Graphics (Integrated)
Jumlah peserta yang bisa ditangani	12	2	12	10

Dari hasil pengujian beberapa peserta diatas, menunjukkan bahwa aplikasi video konferensi dapat menangani hingga 6

peserta dengan kualitas yang cukup baik. Namun, saat jumlah peserta meningkat hingga 8 sampai 12, beberapa penurunan kualitas dan stabilitas koneksi mulai muncul. Kesimpulan hasil pengujian kinerja video konferensi telah diuraikan pada tabel 2 berikut.

Tabel 2. Hasil Pengujian Sistem

No	Pengujian	Hasil
1	Pengujian dengan 2 peserta	Kualitas video dan audio, komunikasi teks baik, latensi rendah dan responsivitas aplikasi baik.
2	Pengujian dengan 3 peserta	Kualitas video dan audio baik, komunikasi teks baik, latensi rendah, responsivitas aplikasi baik.
3	Pengujian dengan 5 peserta	Kualitas video sedikit menurun, audio baik, komunikasi teks baik, latensi video dan audio sedikit <i>delay</i> terutama pada koneksi internet lambat, responsivitas aplikasi cukup baik.
4	Pengujian dengan 6 peserta	Kualitas video dan audio sedikit menurun, terutama pada koneksi internet lambat, latensi video dan audio sedikit <i>delay</i> , komunikasi teks baik, responsivitas aplikasi masih baik.
5	Pengujian dengan 8 peserta	Kualitas video dan audio sedikit menurun, latensi video dan audio sedikit <i>delay</i> , komunikasi teks baik, responsivitas aplikasi menurun, terdapat beberapa gangguan koneksi seperti disconnect dari room meeting.
6	Pengujian dengan 12 peserta	Kualitas video menurun signifikan, audio sedikit gangguan, latensi video dan audio <i>delay</i> cukup lama, komunikasi teks <i>delay</i> , responsivitas aplikasi menurun drastis, beberapa peserta mengalami keterlambatan signifikan.

4. Pengujian Kinerja Fitur Face Recognition  
Aplikasi video konferensi yang dibuat memiliki fitur face recognition untuk

melakukan verifikasi identitas peserta secara otomatis. Proses pengujian kinerja fitur face recognition untuk mengetahui kemampuan aplikasi dalam mengenali beberapa wajah, serta dilakukan dengan jumlah peserta yang berbeda untuk dibandingkan.

Pengujian dilakukan dengan berbagai kondisi, diantaranya menggunakan representasi gambar dari kamera smartphone, peserta menggunakan masker dan penggunaan fileter wajah. Berikut hasil pengujian seperti pada tabel 3 berikut:

Tabel 3. Hasil Pengujian Face Recognition

No	Pengujian	Hasil
1	Pengujian peserta menggunakan representasi gambar	Aplikasi dapat mengenali wajah peserta yang menggunakan gambar representasi dari smartphone selama wajah yang ditampilkan jelas. Informasi peserta dimasukkan ke <i>attending list</i> .
2	Pengujian peserta menggunakan masker	Aplikasi tidak dapat mengenali wajah peserta yang menggunakan masker. Peserta yang tidak menggunakan masker masih dapat dikenali. Informasi peserta yang menggunakan masker tidak dimasukkan ke <i>attending list</i> .
3	Pengujian peserta dengan wajah yang sesuai	Aplikasi dapat mengenali wajah peserta yang sesuai dengan wajah saat membuat akun. Wajah yang sesuai akan muncul dengan bounding box dan informasi nama peserta dimasukkan ke <i>attending list</i> .
4	Pengujian peserta dengan wajah yang tidak sesuai	Aplikasi tidak dapat mengenali wajah peserta yang berbeda dengan wajah saat membuat akun. Informasi peserta tidak dimasukkan ke

		<i>attending list</i> . Pada pengujian dengan beberapa peserta yang bertukar wajah, aplikasi juga tidak mengenali wajah mereka.
5	Pengujian peserta menggunakan filter wajah	Aplikasi tidak dapat mengenali wajah peserta yang menggunakan filter wajah. Informasi peserta yang menggunakan filter tidak dimasukkan ke <i>attending list</i> .

## 5. KESIMPULAN

Pada penelitian ini, pembuatan aplikasi video konferensi dengan nama Attend Meet telah berhasil diselesaikan dengan mengintegrasikan teknologi *face recognition* menggunakan *face-api.js*. Sehingga Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Aplikasi Attend Meet memungkinkan pengguna untuk mengadakan atau mengikuti sesi video konferensi dengan fitur-fitur seperti *live video*, *live microphone*, *live chat*, dan *share screen*. Selain itu, fitur *face recognition* berhasil diintegrasikan menggunakan *face-api.js*.
2. Aplikasi Attend Meet mampu mendeteksi wajah peserta secara otomatis melalui fitur *face recognition* dengan membandingkan data wajah yang telah terdaftar di database dengan wajah yang tampil pada video konferensi. Proses verifikasi ini berjalan secara *realtime* selama fitur *face recognition* diaktifkan, sehingga memastikan bahwa hanya peserta yang sesuai yang di presensi/dihadirkan. Hasil pengujian menunjukkan bahwa fitur *face recognition* yang digunakan mampu mengenali beberapa wajah sekaligus selama kualitas wajah pada video yang ditampilkan jelas.

## DAFTAR PUSTAKA

- [1] Andri Nugraha Ramdhon, & Fadly Febriya. (2021). Penerapan Face Recognition Pada Sistem Presensi. *Journal of Applied Computer Science and Technology*, 2(1), 12–17. <https://doi.org/10.52158/jacost.v2i1.121>
- [2] Arsal, M., Agus, B., & Anggraini, D. (2020). Jurnal Nasional Teknologi dan Sistem Informasi Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning. 01, 55–63. <https://teknosi.fti.unand.ac.id/index.php/teknosi/article/view/1511>
- [3] Fajrin, M. U., & Tiorida, E. (2020). Faktor yang Memengaruhi Minat Perilaku Penggunaan Teknologi (Studi: Pengguna Aplikasi Video Conference selama Physical Distancing). *Prosiding The 11th Industrial Research Workshop and National Seminar*, 977–984. <https://jurnal.polban.ac.id/proceeding/article/view/2151>
- [4] Febrianto Widyoutomo, Hamidillah Ajie, & Widodo. (2021). Pengembangan Web Service Modul Mahasiswa Pada Sistem Informasi Akademik Universitas Negeri Jakarta. *PINTER: Jurnal Pendidikan Teknik Informatika Dan Komputer*, 5(1), 68–75. <https://doi.org/10.21009/pinter.5.1.9>
- [5] Ismawati, D., & Prasetyo, I. (2020). Efektivitas Pembelajaran Menggunakan Video Zoom Cloud Meeting pada Anak Usia Dini Era Pandemi Covid-19. *Jurnal Obsesi: Jurnal Pendidikan Anak Usia Dini*, 5(1), 665. <https://doi.org/10.31004/obsesi.v5i1.671>
- [6] Lim, K. Y., & , Hau Joan, Y. T. (2022). Evaluation Of Virtual Classroom With Artificial Intelligence. 2021(Icdxa 2021), 1–6. <https://iemjournal.com.my/index.php/ie/article/view/116>
- [7] Mubarak, H. (2019). Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN). *Universitas Islam Negeri Maulana Malik Ibrahim Malang*, 3(1), 10–12. <http://etheses.uin-malang.ac.id/15295/>
- [8] Muhamma naim (2022), Rancang Bangun Sistem Deteksi Kehadiran Otomatis pada Media Pembelajaran Video Online Menggunakan Metode Pengenalan Wajah <https://repository.unhas.ac.id/id/eprint/12792/>
- [9] Tanuwijaya, E., Lordianto, R. L., & ... (2022). Recognition of Human Faces in Video Conference Applications Using the Cnn Pipeline. *Jurnal Teknik Informatika* 3(2), 421–427. <http://jutif.if.unsoed.ac.id/index.php/jurnal/article/download/219/80>